

1      **CLAIMS**

2

3      1. One or more electronically-accessible media comprising  
4      electronically-executable instructions that comprise at least part of an operating  
5      system, the at least part of the operating system including:

6                an application programming interface that is capable of creating callback-  
7          type dynamic function tables, each callback-type dynamic function table including  
8          a begin address, an end address, and a callback function; wherein the operating  
9          system calls the callback function when an exception occurs within an address  
10       range bounded by the begin address and the end address, the callback function  
11       requesting exception handling and/or unwind information from a runtime  
12       component.

13

14      2. The one or more electronically-accessible media as recited in claim 1,  
15      wherein the callback function returns from the runtime component data for code  
16      address-to-pointer information.

17

18      3. The one or more electronically-accessible media as recited in claim 2,  
19      wherein the data for code address-to-pointer information comprises a reference to  
20      the code address-to-pointer information.

21

22      4. The one or more electronically-accessible media as recited in claim 2,  
23      wherein the data for code address-to-pointer information comprises the code  
24      address-to-pointer information.

1       5. The one or more electronically-accessible media as recited in claim 4,  
2 wherein the code address-to-pointer information comprises a start address, a final  
3 address, and an unwind pointer.

4

5       6. One or more electronically-accessible media comprising  
6 electronically-executable instructions that utilize an application programming  
7 interface, the application programming interface facilitating creation of callback-  
8 type dynamic function tables; each callback-type dynamic function table including  
9 a begin address, an end address, and a callback function, each callback-type  
10 dynamic function table corresponding to a code heap that stores code for a  
11 plurality of functions in a runtime environment; wherein interaction between the  
12 runtime environment and an operating system is precipitated upon calling the  
13 callback function to acquire exception handling and/or unwind information.

14

15       7. The one or more electronically-accessible media as recited in claim 6,  
16 wherein the interaction between the runtime environment and the operating system  
17 comprises (i) the operating system requesting exception handling and/or unwind  
18 information from the runtime environment and (ii) the runtime environment  
19 providing the exception handling and/or unwind information to the operating  
20 system.

1       **8.**     The one or more electronically-accessible media as recited in claim 6,  
2     wherein the begin address and the end address of each callback-type dynamic  
3     function table comprises a begin address and an end address of a respective code  
4     heap to which a respective callback-type dynamic function table corresponds.

5

6       **9.**     The one or more electronically-accessible media as recited in claim 6,  
7     wherein the exception handling and/or unwind information comprises data for  
8     code address-to-pointer information for a particular function of the plurality of  
9     functions.

10

11      **10.**    The one or more electronically-accessible media as recited in claim  
12     9, wherein the particular function of the plurality of functions is ascertained using  
13     an instruction pointer that is passed as an argument of the callback function.

14

15      **11.**    The one or more electronically-accessible media as recited in claim  
16     6, wherein each callback-type dynamic function table is created, at least in part, by  
17     adding each callback-type dynamic function table to a linked list of dynamic  
18     function tables.

19

20      **12.**    The one or more electronically-accessible media as recited in claim  
21     11, wherein the linked list of dynamic function tables includes one or more non-  
22     callback-type dynamic function tables.

1       **13.** One or more electronically-accessible media comprising  
2 electronically-executable instructions that comprise at least part of a runtime  
3 component, the runtime component adapted to manage exception handling and/or  
4 unwind information for a plurality of functions that are executing as part of a  
5 runtime environment, the runtime component capable of providing data for code  
6 address-to-pointer information to an operating system when an exception occurs  
7 with respect to a particular function of the plurality of functions.

8  
9       **14.** The one or more electronically-accessible media as recited in claim  
10 13, wherein the data for code address-to-pointer information comprises a reference  
11 to the code address-to-pointer information for the particular function of the  
12 plurality of functions.

13  
14       **15.** The one or more electronically-accessible media as recited in claim  
15 13, wherein the data for code address-to-pointer information comprises the code  
16 address-to-pointer information for the particular function of the plurality of  
17 functions.

18  
19       **16.** The one or more electronically-accessible media as recited in claim  
20 15, wherein the code address-to-pointer information comprises a start address, a  
21 final address, and an unwind pointer for the particular function of the plurality of  
22 functions.

**17. An electronic device comprising:**

a runtime environment that is managing code for a plurality of functions;

and

an operating system that is managing a linked list of dynamic function tables that are searched when an exception occurs, the operating system adapted to call a callback function as indicated by a dynamic function table of the linked list of dynamic function tables to request that the runtime environment provide exception handling and/or unwind information for at least one function of the plurality of functions;

wherein the runtime environment is capable of providing to the operating system the exception handling and/or unwind information for the at least one function of the plurality of functions responsive to the callback function.

**18.** The electronic device as recited in claim 17, wherein the dynamic function table comprises a callback dynamic function table.

**19.** The electronic device as recited in claim 18, wherein the linked list of dynamic function tables comprises at least one sorted dynamic function table and at least one unsorted dynamic function table.

**20.** The electronic device as recited in claim 17, wherein the dynamic function table corresponds to more than one function of the plurality of functions.

1       **21.** The electronic device as recited in claim 17, wherein the dynamic  
2 function table includes a begin address entry and an end address entry that reflect a  
3 begin address and an end address, respectively, of a code heap that includes more  
4 than one function of the plurality of functions, including the at least one function  
5 of the plurality of functions.

6

7       **22.** The electronic device as recited in claim 17, wherein the runtime  
8 environment comprises at least one just-in-time (JIT) compiler.

9

10      **23.** The electronic device as recited in claim 17, wherein the exception  
11 handling and/or unwind information comprises data for code address-to-pointer  
12 information for the at least one function of the plurality of functions.

13

14      **24.** The electronic device as recited in claim 23, wherein the data for  
15 code address-to-pointer information for the at least one function of the plurality of  
16 functions directly or indirectly provides to the operating system a start address, a  
17 final address, and an unwind pointer for the at least one function of the plurality of  
18 functions.

25. One or more electronically-accessible media comprising a data structure, the data structure comprising:

a begin address;

an end address; and

a callback function that, when called, returns from a runtime environment exception handling and/or unwind information for a function associated with at least one address that is between the begin address and the end address.

26. The one or more electronically-accessible media as recited in claim  
herein the begin address and the end address reflect a begin address and an  
address, respectively, for a code heap of the runtime environment.

**27.** The one or more electronically-accessible media as recited in claim 26, wherein the code heap includes code for a plurality of functions.

**28.** The one or more electronically-accessible media as recited in claim 27, wherein the callback function, when called, is capable of returning from the runtime environment exception handling and/or unwind information for any function of the plurality of functions.

29. The one or more electronically-accessible media as recited in claim 25, wherein the callback function is called by an operating system to enable the operating system to unwind a stack.

1       **30.**   The one or more electronically-accessible media as recited in claim  
2 25, wherein the callback function accepts as input the at least one address that is  
3 associated with the function, the at least one address comprising an instruction  
4 pointer.

5

6       **31.**   The one or more electronically-accessible media as recited in claim  
7 25, wherein the callback function accepts as input a reference to at least part of the  
8 data structure.

9

10      **32.**   The one or more electronically-accessible media as recited in claim  
11 25, wherein the exception handling and/or unwind information comprises data for  
12 code address-to-pointer information for the function that is associated with the at  
13 least one address.

14

15      **33.**   The one or more electronically-accessible media as recited in claim  
16 25, wherein the begin address, the end address, and the callback function together  
17 comprise a callback dynamic function table.

18

19      **34.**   The one or more electronically-accessible media as recited in claim  
20 33, wherein the data structure further comprises:

21            a plurality of dynamic function tables, the plurality of dynamic function  
22 tables including the callback dynamic function table.

35. An electronic device comprising:

at least one processor; and

one or more media in operative communication with the at least one processor, the one or more media including a data structure comprising:

- a begin address value;
- an end address value; and
- a callback function that, when called, returns from a runtime environment exception handling and/or unwind information for a function associated with at least one address that is between the begin address value and the end address value.

**36.** The electronic device as recited in claim 35, wherein the one or more media further include a code heap managed by the runtime environment, the code heap having a begin address and an end address; the begin address value and the end address value reflecting the begin address and the end address, respectively, of the code heap.

**37.** The electronic device as recited in claim 36, wherein the code heap includes code for a plurality of functions.

**38.** The electronic device as recited in claim 37, wherein the callback function, when called, is capable of returning from the runtime environment exception handling and/or unwind information for any function of the plurality of functions responsive to the at least one address.

1           **39.** The electronic device as recited in claim 35, wherein the one or  
2 more media further include a stack; and wherein the callback function is called by  
3 an operating system to enable the operating system to unwind the stack.

4

5           **40.** One or more electronically-accessible media comprising  
6 electronically-executable instructions that include:

7                 a callback function, the callback function accepting as input an instruction  
8 pointer that is associated with an address of a function from a runtime  
9 environment and producing as output data for code address-to-pointer information  
10 for the function having the address that is associated with the instruction pointer;

11                 wherein the callback function may be called by an operating system and  
12 implemented by the runtime environment.

13

14           **41.** The one or more electronically-accessible media as recited in claim  
15 40, wherein the callback function further accepts as input a reference to a callback  
16 dynamic function table for context.

17

18           **42.** The one or more electronically-accessible media as recited in claim  
19 41, wherein the callback dynamic function table includes a begin address and an  
20 end address that define an address range that includes the instruction pointer.

1           **43.**   The one or more electronically-accessible media as recited in claim  
2   40, wherein the data for code address-to-pointer information that is output by the  
3   callback function is provided from the runtime environment to the operating  
4   system.

5

6           **44.**   The one or more electronically-accessible media as recited in claim  
7   40, wherein the data for code address-to-pointer information comprises a reference  
8   to the code address-to-pointer information.

9

10          **45.**   The one or more electronically-accessible media as recited in claim  
11   44, wherein the referenced code address-to-pointer information comprises a start  
12   address, a final address, and an unwind pointer for the function having the address  
13   that is associated with the instruction pointer.

14

15          **46.**   The one or more electronically-accessible media as recited in claim  
16   40, wherein the data for code address-to-pointer information comprises the code  
17   address-to-pointer information.

18

19          **47.**   The one or more electronically-accessible media as recited in claim  
20   40, wherein the callback function is adapted to be called by the operating system  
21   during an exception handling procedure.

1       **48.**   The one or more electronically-accessible media as recited in claim  
2       40, wherein the callback function is implemented, at least partially, by the runtime  
3       environment by inspecting a code header for the function having the address that  
4       is associated with the instruction pointer.

5

6       **49.**   The one or more electronically-accessible media as recited in claim  
7       48, wherein the data for code address-to-pointer information is derived using the  
8       code header.

9

10      **50.**   The one or more electronically-accessible media as recited in claim  
11     48, wherein the code header is ascertained using a heap structure contents of a  
12     code heap that includes code for the function having the address that is associated  
13     with the instruction pointer.

14

15      **51.**   One or more electronically-accessible media comprising at least part  
16     of an operating system that is configured to request from a runtime environment  
17     exception handling and/or unwinding information for functions that are managed  
18     by the runtime environment.

19

20      **52.**   The one or more electronically-accessible media as recited in claim  
21     51, wherein the at least part of the operating system is further configured to  
22     request from the runtime environment the exception handling and/or unwinding  
23     information after an exception is discovered.

1       **53.** The one or more electronically-accessible media as recited in claim  
2 51, wherein the at least part of the operating system is further configured to  
3 request the exception handling and/or unwinding information from the runtime  
4 environment responsive to locating a dynamic function table having a callback  
5 function.

6

7       **54.** The one or more electronically-accessible media as recited in claim  
8 53, wherein the dynamic function table having the callback function corresponds  
9 to at least two functions compiled in the runtime environment by a just-in-time  
10 (JIT) compiler.

11

12     **55.** An electronic device comprising:

13       at least one processor; and  
14       one or more media including processor-executable instructions that are  
15 capable of being executed by the at least one processor, the processor-executable  
16 instructions adapted to cause the electronic device to perform actions comprising:

17           initializing a code heap having a begin address and an end address  
18 for a runtime; and

19           creating a dynamic function table corresponding to the code heap,  
20 the dynamic function table including the begin address, the end address,  
21 and a callback function.

1       **56.** The electronic device as recited in claim 55, wherein the initializing  
2 action comprises:

3                     initializing the code heap by at least one of the runtime and an  
4 operating system.

5  
6       **57.** The electronic device as recited in claim 55, wherein the processor-  
7 executable instructions are adapted to cause the electronic device to perform a  
8 further action comprising:

9                     installing the dynamic function table as part of a linked list of  
10 dynamic function tables.

11  
12      **58.** The electronic device as recited in claim 57, wherein the processor-  
13 executable instructions are adapted to cause the electronic device to perform a  
14 further action comprising:

15                     removing the dynamic function table from the linked list of dynamic  
16 function tables.

17  
18      **59.** The electronic device as recited in claim 55, wherein the processor-  
19 executable instructions are adapted to cause the electronic device to perform  
20 further actions comprising:

21                     initiating the callback function by an operating system responsive to  
22 an exception; and

23                     providing data for code address-to-pointer information from the  
24 runtime to the operating system.

1       **60.** The electronic device as recited in claim 55, wherein the processor-  
2 executable instructions are adapted to cause the electronic device to perform a  
3 further action comprising:

4              storing, by the runtime, code for a plurality of functions in the code  
5 heap.

6

7       **61.** One or more electronically-accessible media comprising  
8 electronically-executable instructions that, when executed, direct an electronic  
9 device to perform actions comprising:

10             initializing a code heap for a runtime, the code heap having a begin address  
11 and an end address; and

12             creating a dynamic function table corresponding to the code heap, the  
13 dynamic function table including the begin address, the end address, and a  
14 callback function.

15

16

17

18

19

20

21

22

23

24

25

1       **62.** One or more electronically-accessible media comprising  
2 electronically-executable instructions that, when executed, direct an electronic  
3 device to perform actions comprising:

4             receiving at a runtime an instruction pointer with a request for data for code  
5 address-to-pointer information;

6             ascertaining a runtime function associated with the instruction pointer;

7             accessing a code header of the ascertained runtime function to extract data  
8 for code address-to-pointer information for the ascertained runtime function; and

9             providing the extracted data for code address-to-pointer information from  
10 the runtime.

11  
12       **63.** The one or more electronically-accessible media comprising the  
13 electronically-executable instructions that, when executed, direct an electronic  
14 device to perform the actions as recited in claim 62, wherein the action of  
15 providing comprises the action of:

16             providing from the runtime a pointer to the code address-to-pointer  
17 information for the ascertained runtime function.

18  
19       **64.** The one or more electronically-accessible media comprising the  
20 electronically-executable instructions that, when executed, direct an electronic  
21 device to perform the actions as recited in claim 62, wherein the action of  
22 providing comprises the action of:

23             providing from the runtime to an operating system the extracted data  
24 for code address-to-pointer information.

1           **65.** An electronic device comprising:

2           at least one processor; and

3           one or more media including processor-executable instructions that are  
4 capable of being executed by the at least one processor, the processor-executable  
5 instructions adapted to cause the electronic device to perform actions comprising:

6           locating a callback dynamic function table having a callback  
7           function;

8           initiating, by an operating system, the callback function to interact  
9           with a runtime;

10          receiving data for code address-to-pointer information from the  
11          runtime responsive to the initiating; and

12          using the received data for code address-to-pointer information to  
13          attain information for at least one of exception handling and stack  
14          unwinding.

15  
16          **66.** The electronic device as recited in claim 65, wherein:

17          the receiving action comprises:

18           receiving a reference to code address-to-pointer information  
19           from the runtime; and

20          the using action comprises:

21           using the reference to attain code address-to-pointer  
22           information, including an unwind pointer; and

23           using the unwind pointer to attain unwind information.

1       **67.** The electronic device as recited in claim 65, wherein the processor-  
2 executable instructions are adapted to cause the electronic device to perform a  
3 further action comprising:

4                  discovering an exception from a runtime function.  
5

6       **68.** The electronic device as recited in claim 65, wherein the processor-  
7 executable instructions are adapted to cause the electronic device to perform a  
8 further action comprising:

9                  searching a dynamic function table linked list using an instruction  
10 pointer;

11                  wherein the locating action comprises:

12                  locating the callback dynamic function table having the  
13 callback function from the dynamic function table linked list using  
14 the instruction pointer, the callback dynamic function table including  
15 a begin address that is less than and an end address that is greater  
16 than the instruction pointer.  
17  
18  
19  
20  
21  
22  
23  
24  
25

**69.** The electronic device as recited in claim 68, wherein the processor-executable instructions are adapted to cause the electronic device to perform further actions comprising:

receiving at the runtime, responsive to the initiating, the instruction pointer with a request for the data for code address-to-pointer information;

ascertaining a runtime function associated with the instruction pointer;

accessing a code header of the ascertained runtime function to extract the data for code address-to-pointer information for the ascertained runtime function; and

providing, from the runtime to the operating system, the data for code address-to-pointer information.

70. One or more electronically-accessible media comprising electronically-executable instructions that, when executed, direct an electronic device to perform actions comprising:

locating a callback dynamic function table having a callback function;

initiating, by an operating system, the callback function to interact with a runtime;

receiving, at the operating system, data for code address-to-pointer information from the runtime responsive to the initiating; and

using the received data for code address-to-pointer information to attain information for at least one of exception handling and stack unwinding.

1       **71.** An arrangement comprising:

2           means for initializing a code heap for a runtime, the code heap having a  
3 begin address and an end address; and

4           means for creating a dynamic function table corresponding to the code  
5 heap, the dynamic function table including the begin address, the end address, and  
6 a callback function.

7       **72.** An arrangement comprising:

8           means for locating a dynamic function table having a callback function;

9           means for initiating the callback function to interact with a runtime  
10 environment;

11           means for receiving data for code address-to-pointer information from the  
12 runtime environment responsive to the callback function initiation; and

13           means for using the received data for code address-to-pointer information  
14 to attain information for at least one of exception handling and stack unwinding.

16

17

18

19

20

21

22

23

24

25